FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Der neue massiv-parallele Rechner CRAY T3E im Frühling 1996 - Erfahrungen mit der Jungfräulichkeit -

Wolfgang E. Nagel

KFA-ZAM-IB-9630

Oktober 1996
(Stand 01.10.96)

Hans-Werner Meuer (Hrsg): Supercomputer 1996,
Anwendungen, Architekturen, Trends
K. G. Saur, München, 1996, pp. 92-107

Der neue massiv-parallele Rechner CRAY T3E im Frühling 1996 - Erfahrungen mit der Jungfräulichkeit -

Wolfgang E. Nagel

Zentralinstitut für Angewandte Mathematik
Forschungszentrum Jülich GmbH (KFA)
E-mail: w.nagel@kfa-juelich.de

Kurzfassung. Nachdem die CRAY T3D nach der Markteinführung ab Mitte 1993 mit mehr als dreißig ausgelieferten Systemen beachtliche Erfolge erzielt und mit Hardware-Eigenschaften (globaler Adreßraum mit schnellem Zugriff auf entfernte Speicherstellen, leistungsfähiges 3D-Netzwerk, schnelle Synchronisation) und Software-Komponenten (*Totalview* als Debugger, *Apprentice* als Performance-Analyse-Werkzeug) Maßstäbe gesetzt hat, ist der Erwartungsdruck auf dem Ende 1995 angekündigten Nachfolgesystem CRAY T3E erheblich. Auch dieser Rechner basiert auf einem Alpha-Chip der Firma DEC (EV5 (21164), 600 MFLOPS Knotenleistung), dessen Speicherzugriffverhalten durch einen auf dem Chip integrierten *secondary cache* verbessert worden ist. Durch unterstützende Hardware-Maßnahmen soll - neben der Erhöhung der Kommunikationsbandbreite des Netzwerkes zwischen je zwei Knoten auf 500 MByte/s und den *E*-Registern zum verbesserten Zugriff auf entfernte Speicherstellen - der Zugriff auf den lokalen Speicher durch sogenannte *stream buffer* beschleunigt werden. Mit dem neuen *GigaRing* steht daneben sowohl für den Anschluß der I/O-Peripherie als auch für die Kommunikation mit anderen Rechnern in einem Cray-Systemkomplex (z.B. einer CRAY T90 oder J90) ein leistungsfähiges Konzept zur Verfügung.

Da zum Zeitpunkt der Drucklegung dieses Beitrages nur einige wenige Prototypen der CRAY T3E mit vorläufigen Hardware-Eigenschaften (Pass 1 Chips) existieren und zudem ein unmittelbarer Zugriff auf ein derartiges - jungfräuliches - System nicht möglich ist, können hier lediglich die wichtigsten Architektureigenschaften und Software-Komponenten sowie die Konzepte zur Einbettung dieses Systems in die Rechenzentrumsumgebung der KFA beschrieben werden. Der Vortrag wird hingegen einen Überblick über die ersten Erfahrungen mit der CRAY T3E geben und anhand von Software-Aspekten im System- und Anwendungsbereich sowie ausgewählten Performance-Daten für Programmkerne und Anwendungen aufzeigen, wie sich dieser neue Rechner vom Vorgängermodell CRAY T3D unterscheidet.

1 Einleitung

Die Lösung komplexer Aufgaben aus Wissenschaft und Technik - wie zum Beispiel aus der Strömungsmechanik bei der Entwicklung neuer Flugzeuge, in der Materialforschung mit dem Ziel zukünftiger Informationstechnologien oder auch bei der Klimamodellierung als einem Schwerpunkt der Umweltforschung - ist wegen des enormen Rechenzeitbedarfs der numerischen Verfahren und mathematischen Simulationen ohne Hochleistungsrechner unmöglich [1]. Seit einigen Jahren drängen Parallelrechner in den Markt, die das Leistungsspektrum bisheriger Rechner erweitern und zudem aus Kostengründen günstiger erscheinen: Durch die Vernetzung von Prozessoren mit preiswerten Chip-Komponenten können nun massiv-parallele Systeme mit verteiltem Speicher realisiert werden, die potentiell eine erheblich höhere Spitzenleistung bieten.

Nachdem die CRAY T3D - als erste Stufe eines dreiphasigen Realisierungskonzeptes der massiv-parallelen Cray-Architekturlinie - nach der Markteinführung ab Mitte 1993 mit mehr als dreißig ausgelieferten Systemen beachtliche Erfolge erzielt und mit Hardware-Eigenschaften (globaler Adreßraum mit schnellem Zugriff auf entfernte Speicherstellen, leistungsfähiges 3D-Netzwerk, schnelle Synchronisation) und Software-Komponenten (*Totalview* als Debugger, *Apprentice* als Performance-Analyse-Werkzeug) Maßstäbe gesetzt hat, ist der Erwartungsdruck auf dem Ende 1995 angekündigten Nachfolgesystem CRAY T3E erheblich. Leider ist zum Zeitpunkt der Drucklegung ein Zugriff auf ein derartiges - noch jungfräuliches - System nicht möglich. Es ist aber geplant, beim Vortrag über aktuelle und praktische Erfahrungen mit diesem Rechner zu berichten.

Das Zentralinstitut für Angewandte Mathematik (ZAM) des Forschungszentrums Jülich (KFA) betreibt seit mehr als 10 Jahren Parallelrechner und stellt seit 1987 einen Teil der Rechenleistung - im Rahmen des Höchstleistungsrechenzentrums HLRZ - mehr als 200 Benutzergruppen zur Verfügung, die über ganz Deutschland verteilt sind und diese zentralen Parallelrechnerressourcen nutzen [2]. Voraussichtlich im Juli 1996 wird das ZAM ein System CRAY T3E mit 512 Prozessoren installieren, und die dann zur Verfügung stehende Rechenleistung wird wiederum zu einem großen Teil von dieser Benutzergemeinschaft zur Lösung von Forschungs- und Entwicklungsaufgaben eingesetzt werden können.

2 Systemarchitektur

Auch das System CRAY T3E folgt der 1993 von Cray für die massiv-parallele Architekturlinie vorgestellten *Makroarchitektur*, die durch die folgenden Eigenschaften gekennzeichnet ist [3]:

- physikalisch verteilter, jedoch global adressierbarer Speicher
- MIMD-Programmiermodell mit effizienter Emulation von SIMD-Primitiven
- Verbindungsnetzwerk mit hoher Bandbreite und niedriger Latenz

Die Aufteilung in eine *Makroarchitektur*, die die Sicht des Anwenders bestimmt und über die Rechner-Generationswechsel hinweg konstant bleibt, und eine

Mikroarchitektur, die sich den jeweiligen technischen Entwicklungen und ökonomischen Randbedingungen anpaßt, ist sinnvoll und notwendig, da sie die bereits geleisteten Investitionen in die Programmentwicklung schützt. Entsprechend dieser Leitlinie konnten die einzelnen Hardware-Komponenten des T3E-Systems - ohne negative Seiteneffekte für die Programmierung - den neuen technischen Möglichkeiten angepaßt werden.

2.1 Das Verbindungsnetzwerk

Die charakteristische Verbindungsstruktur der CRAY T3E ist weiterhin der 3-dimensionale Torus (Abb. 1), der auch für Systeme mit hohen Prozessorzahlen kurze Kommunikationswege bietet (maximale Länge bei einem System mit 512 Knoten: 12 Hops). Die Kommunikationsbandbreite des Netzwerkes zwischen je zwei Knoten konnte allerdings auf 500 MByte/s (*pay load:* bis zu 480 MByte/s) angehoben werden [3].

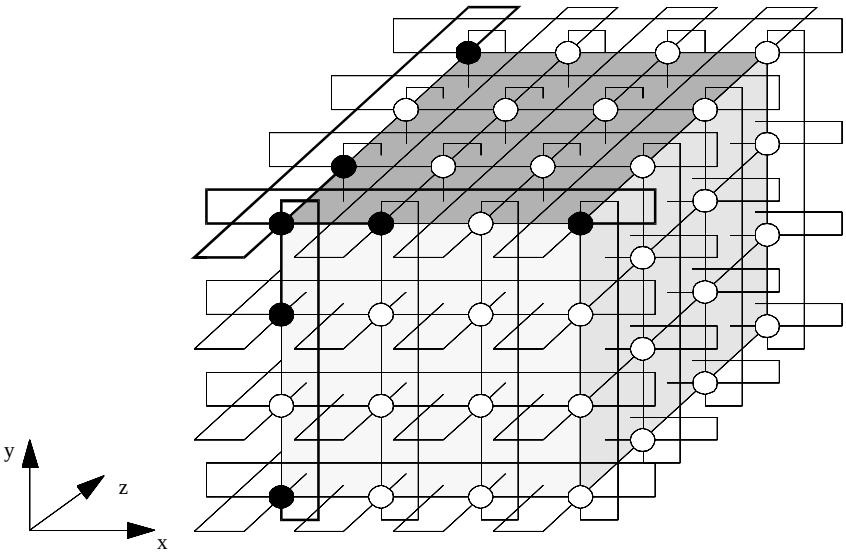

Abbildung 1
3-dimensionale Torus-Struktur der CRAY T3E [4]

Dies bedeutet im Vergleich zur T3D eine Steigerung um den Faktor 6, wenn man berücksichtigt, daß nun von einem Netzwerkknoten nur noch ein Rechenknoten unterstützt werden muß (Abb. 2). Durch die dreidimensionale Kommunikationsstruktur ergibt sich damit eine maximale Durchsatzbandbreite von fast 3 GByte/s pro

Netzwerkknoten, da die Kommunikationsobjekte gleichzeitig bidirektional in jede der drei Richtungen *x*, *y* und *z* transportiert werden können.

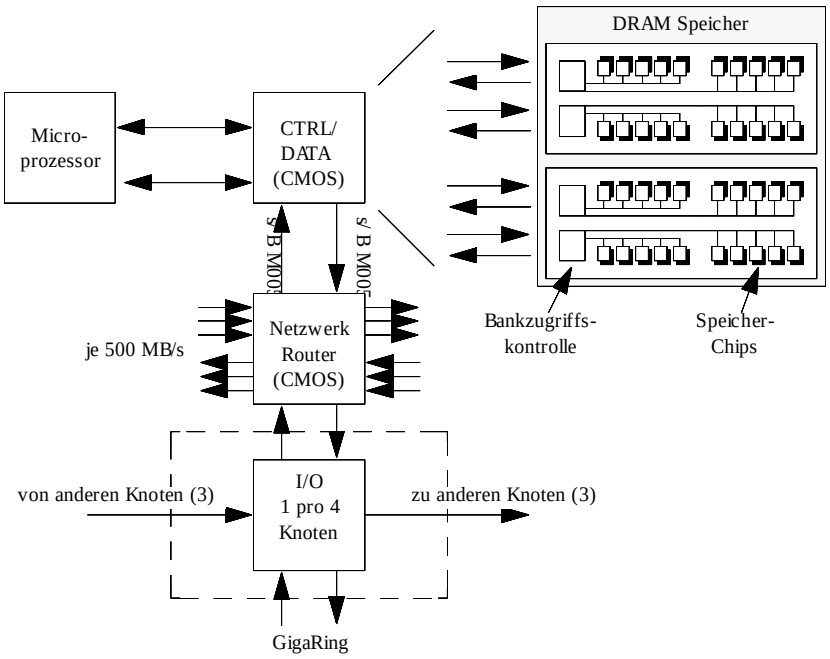

Abbildung 2
Strukturschaltbild von einem Knoten eines T3E-Systems [4]

2.2 Der Prozessor

Auch das T3E-System basiert auf einem Alpha-Chip der Firma DEC: Im Gegensatz zum EV4 beim T3D mit einer Spitzenleistung von 150 MFLOPS wird nun jedoch der mit 300 MHz getaktete EV5 (21164) verwendet, der 4 unabhängige Pipelines (2 Floating-Point, 2 Integer/Logic) besitzt und damit bis zu 1200 MIPS oder 600 MFLOPS liefert. Die Caches für die Instruktionen und Daten blieben sowohl strukturell als auch im Hinblick auf ihre Größe (8 KByte) unverändert, die Speicherbandbreite des Prozessors beträgt nun 1.2 GByte/s. Durch die Integration eines *secondary cache* auf dem Chip mit einer Kapazität von 96 KByte (3-fach assoziativ) konnte das Speicherverhalten des Chips weiter verbessert werden.

2.3 Die Speicherorganisation

Die Speicherorganisation ist in nahezu allen Fällen der bestimmende Faktor für die erzielbare Prozessorleistung. Während in der Vergangenheit bei Parallelrechnern schon die zur Verfügung stehende Speichergröße der limitierende Faktor war

(Maximum auf T3E: 512 MByte je Knoten), sind heute - bei Berücksichtigung der gestiegenen Taktraten und der durch die DRAM-Technologie vorgegebenen übrigen Randbedingungen - leistungsfähige Strukturkonzepte zur Beschleunigung des lokalen Speicherzugriffs bestimmend für den Erfolg einer Parallelrechnerarchitektur. Neben der Verschränkung des Speichers in acht Bänke - die auf der Ebene von 64-Bit-Worten vorgenommen wird und damit den quasi-parallelen Zugriff von *cache lines* ermöglicht - soll der Zugriff auf den lokalen Speicher durch sogenannte *stream buffer* beschleunigt werden (Abb. 3).

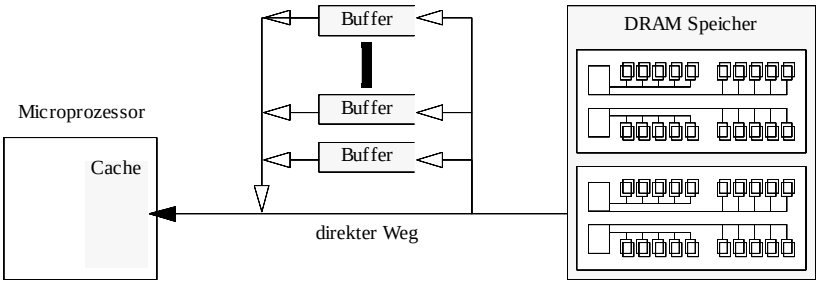

Abbildung 3
Einsatz der *stream buffer* zur Optimierung des Speicherzugriffs [4]

Da die Realisierung von vielen großen externen Caches für jeden der Prozessoren in einem massiv-parallelen Rechnersystem zu aufwendig ist, kommen im T3E-Rechner *stream buffer* zum Einsatz, die konsekutive Datenzugriffe auf den Speicher - auch wenn sie zeitlich nicht unmittelbar aufeinander folgen - durch die Hardware dynamisch entdecken. Durch *Prefetch*-Maßnahmen lassen sich - für geeignete Programme mit vielen Vektor- oder Feldzugriffen - mit diesem Konzept erhebliche Leistungsverbesserungen erzielen [4].

2.4 Realisation des globalen Adreßraumes

Die im Vergleich zu anderen massiv-parallelen Systemen hervorstechende Besonderheit des T3E-Rechners ist die Bereitstellung eines globalen Adreßraumes. Der Zugriff auf globale Daten, die auf einem anderen Prozessor gespeichert sind, wird implizit über die *E*-Register realisiert: Jeder der Netzwerkknoten verfügt über einen in schneller SRAM-Technologie ausgeführten *E*-Registersatz mit 512 Registern, die als Schnittstelle für den entfernten Speicherzugriff dienen; daneben gibt es weitere 128 *E*-Register, die für spezielle Systemfunktionen reserviert sind (siehe Abb. 4).
Der Zugriff aus dem Mikroprozessor auf die *E*-Register erfolgt - am *cache* vorbei - über *load/store*-Operationen. Der Prozessor spezifiziert einen Zielbereich für eine *get/put*-Speicheroperation in einem *E*-Register, und der Netzwerk-Router transferiert - asynchron und parallel zu den weiteren Prozessor-Operationen - die gewünschten Daten (8 oder 64 Byte) an die gewünschte Stelle. Dabei werden auch bei dem

korrespondierenden Prozessor die entsprechenden *E*-Register verwendet (siehe Abb. 4, [4]), die *cache*-Kohärenz wird durch eine zusätzliche Invalidierungs-Komponente sichergestellt. Da die *get*-Operationen spekulativ abgesetzt werden können, sollen sich Latenzzeiten, die bei einem entfernten Zugriff zunächst unvermeidlich sind, in vielen Fällen durch diese Hardware-Unterstützung verstecken oder zumindest mildern lassen.

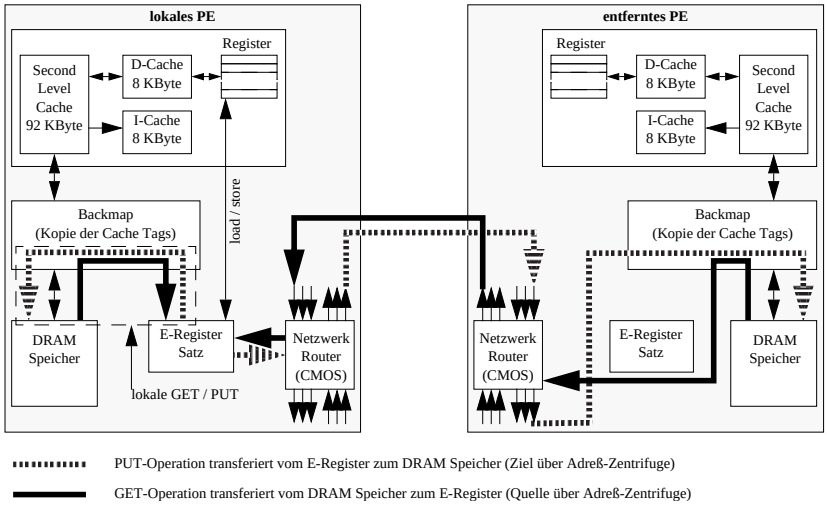

Abbildung 4
Nutzung der *E*-Register für *PUT/GET*-Operationen [4]

2.5 I/O-Konzeption der T3E

Typisches Merkmal für die Bearbeitung von realen Anwendungen ist die Ein- und Ausgabe von großen Datenmengen; obwohl diese Tatsache unumstritten und seit vielen Jahren bekannt ist, war das I/O-Verhalten bei bisherigen Parallelrechnern immer noch einer der Flaschenhälse, der die effiziente parallele Behandlung des Problems häufig behindert und manchmal sogar unmöglich gemacht hat. Mit der neuen I/O-Konzeption der T3E (siehe Abb. 5) soll dieses Problem auch für hochparallele Systeme der Vergangenheit angehören: Mit der Zuordnung von jeweils vier Rechenknoten zu einem I/O-Knoten (Transferleistung max. jeweils 500 MByte/s) und der Anbindung der Peripherie über den auf der Basis des *SCI*-Standards neu entwickelten *GigaRing* (max. Bandbreite: 800 Mbyte/s bidirektional) wird die I/O-Leistung weitgehend von der angeschlossenen Peripherie begrenzt (siehe Abb. 6).

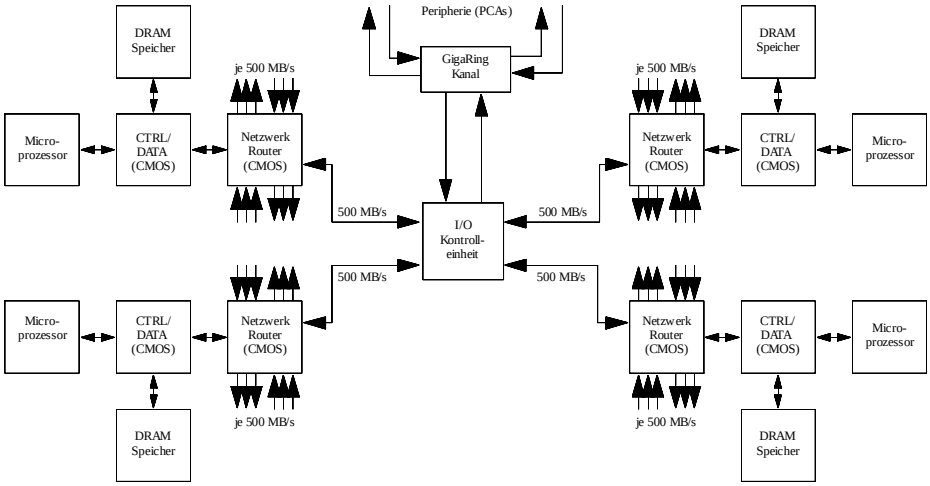

Abbildung 5
Organisation und Bandbreiten der Peripherie-Anbindung [4]

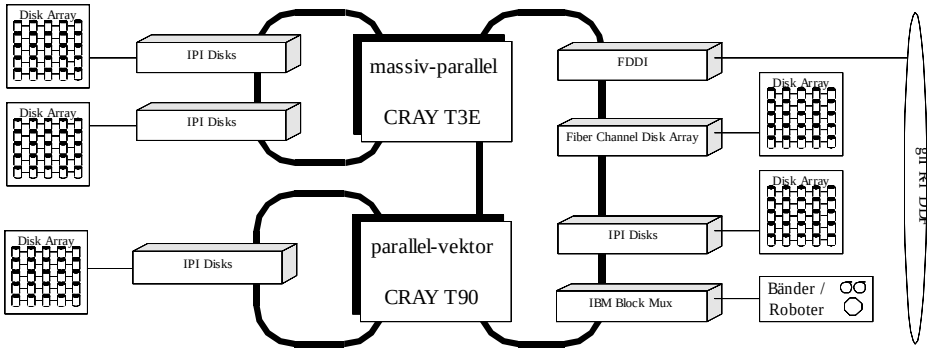

Abbildung 6
Schnelle Kopplung von Vektor- und massiv-paralleler Rechnerkomponente mit einer GigaRing-Infrastruktur [4]

3 Programmierung und Software-Produkte

Die Verfügbarkeit von leistungsfähiger Software bestimmt in den letzten Jahren auch bei Parallelrechnern mehr und mehr die Wahl der Rechnerplattform. Neben Compilern für die gängigen Programmiersprachen (im technisch-wissenschaftlichen Bereich Fortran, C und in jüngster Zeit mehr und mehr auch C++), effizienten Schnittstellen für das heute sicher noch unvermeidliche Message Passing (genannt seien hier MPI, PVM und die von Cray bereitgestellten *shmem*-Routinen zur

einseitigen Kommunikation) und unterstützenden Programmierwerkzeugen (*Totalview* als Debugger, *Apprentice* als Performance-Analyse-Werkzeug) ist häufig die Verfügbarkeit von wichtigen Programmpaketen für die unterschiedlichen Anwendungsgebiete ein gewichtiges Entscheidungskriterium; für den Bereich der Chemie werden hier exemplarisch *GAMESS, GAUSSIAN, MOLPRO* und *UNICHEM* genannt. In [3] sind ca. 30 Pakete aus dem industriellen Anwendungsbereich angesprochen, für die eine Portierung bereits angestoßen ist und in 1996 abgeschlossen werden soll.
Neben diesem sehr wichtigen Bereich sei hier noch einmal erwähnt, daß die Entwicklung und Bereitstellung von leistungsfähigen und effizient implementierten Programmiermodellen, die dem Anwender den globalen Adreßraum als virtuell gemeinsamen Adreßraum unmittelbar zur Verfügung stellen, langfristig für den kommerziellen Erfolg von Parallelrechnern unabdingbar erscheint [1, 5]. Hier kann das Entwicklungsteam der Firma Cray auf die mit dem T3D-Produkt *CRAy ForTran* (*CRAFT [6]*) gemachten Erfahrungen aufbauen, und es bleibt zu hoffen, daß das für Ende 1996 angekündigte neue Produkt *HPF-CRAFT* [7] die wichtigen und bei der Programmierung nutzbringenden Eigenschaften und Sprachprimitive von *CRAFT* effizient einbindet.

4 Einbettung der CRAY T3E in eine Rechenzentrumsumgebung

Trotz der außerordentlichen Erfolge, die in einigen Anwendungsprojekten durch den Einsatz von Parallelrechnern bisher erzielt werden konnten und die die weiteren Forschungsanstrengungen beflügeln, ist in der Praxis für ein breiteres Spektrum von Anwendungen der breite Durchbruch der Parallelverarbeitung bis heute noch nicht erzielt worden [1]. Es hat sich gezeigt, daß die Bereitstellung von Rechenleistung und eine ausreichende Software-Funktionalität auf dem Parallelrechner allein nicht ausreichen, um die verschiedenen DV-Aufgaben in einem wissenschaftlich-technischen Umfeld anzugehen: Auch auf alle anderen Komponenten eines Rechenzentrums kommen beim Betrieb von derartigen Rechnern sehr hohe Anforderungen zu, die zunächst erfüllt werden müssen, bevor die Projekte erfolgreich bearbeitet werden können [1, 2].

4.1 Einbindung in die Infrastruktur

Die effektive Nutzung von Hochleistungsrechnern hängt ganz wesentlich von der Einbindung in die übrige Infrastruktur ab. Neben dem Wunsch nach leichtem Zugriff auf sehr große Speicherkapazitäten für Projektdaten - eine der klassischen Aufgaben in Rechenzentren und meist auch eine unabdingbare Voraussetzung für die erfolgreiche Bewältigung von Großprojekten - steht heute als zusätzliche Anforderung immer häufiger die leistungsfähige Visualisierung des aktuellen Berechnungszustandes auf entfernt stehenden Workstations mit Hochleistungs-graphik im Mittelpunkt, einhergehend mit der Möglichkeit der interaktiven Steuerung von Simulationsläufen. Damit erhält die Anbindung der Rechnerressourcen an die Kommunikationsnetze auf allen beteiligten Ebenen eine

zentrale Bedeutung (Anbindung der lokalen Workstations, schnelle Rechner-Rechner-Kopplung, effiziente *Wide Area Network (WAN)*-Anbindung - zum Beispiel auf der Basis von ATM - zur schnellen Visualisierung und zum Datentransfer).

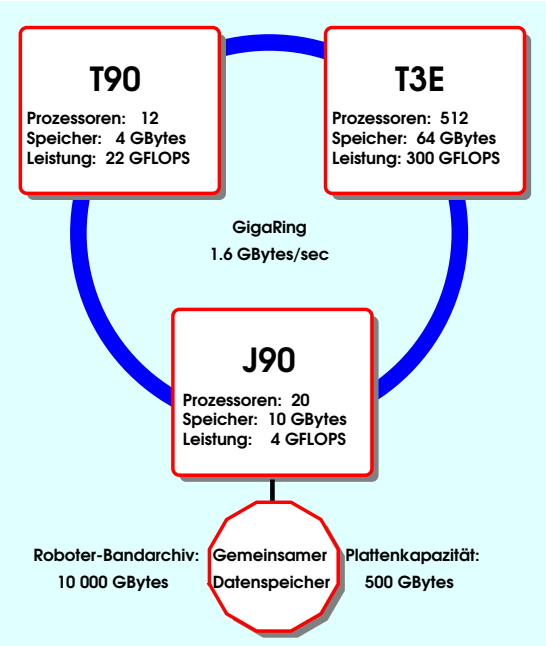

Abbildung 7
Zielkonfiguration der CRAY-Systeme im Forschungszentrum Jülich

Diese Anforderungen wurden bei der Planung der Zielkonfiguration für einen Hochleistungsrechnerkomplex im Forschungszentrum Jülich mit einbezogen (siehe Abb. 7). Mit einem Rechner für interaktive Arbeiten (z.B. Visualisierung, Debugging etc.), kleinen *Batch*-Anwendungen und Fileserver-Diensten (CRAY J90 mit 20 Prozessoren), einem leistungsfähigen Vektor-Supercomputer (CRAY T90 mit 12 Prozessoren) und der hier im Mittelpunkt stehenden massiv-parallelen Komponente CRAY T3E mit 512 Knoten können - unter Nutzung der extrem schnellen GigaRing-Hardware zum Austausch von Daten zwischen diesen Komponenten - Randbedingungen geschaffen werden, die für die zukünftige Nutzung von Hochleistungsrechnern in vielen Bereichen unabdingbar sind. Die Außenanbindung auf der Basis der im ZAM bereits realisierten HiPPI- und ATM-Infrastruktur macht dann diese Ressourcen für die Nutzergemeinde sowohl auf dem Campus als auch bundesweit verfügbar.

4.2 Anpassung an produktionsorientierte Betriebsabläufe

In der Vergangenheit wurden Parallelrechner überwiegend im Kontext von universitären Forschungsvorhaben eingesetzt, bei denen einige wenige Anwender im lokalen Umfeld des Rechnerstandortes den Betrieb und die Nutzung der Rechnerressourcen untereinander absprechen konnten. Mit der seit zwei bis drei Jahren zu beobachtenden Integration von Parallelrechnern in Produktionsumgebungen [1, 8] ergibt sich eine Vielzahl von neuen Problemen, die durch die Betriebs-Software dieser Systeme in den ersten Software-Versionen häufig nicht abgedeckt werden. Über die Koordination des Rechnerzugangs hinaus müssen auch für Parallelrechner typische Rechenzentrumsdienste wie die globale Ressourcenverwaltung und das Accounting angeboten werden, um die Projektabwicklung geeignet sicherzustellen.
Dabei gewinnt das effiziente Scheduling von parallelen Anwendungen [9, 10] zur Steigerung der Auslastung mehr und mehr an Bedeutung, und erste Realisierungen - sogenannte *politische Scheduler* - für parallele Systeme werden unter Lastbedingungen erst zeigen müssen, inwieweit die Notwendigkeiten des täglichen Betriebes durch die bisher entwickelten Konzepte abgedeckt werden können. Ähnliches gilt für das *Checkpointing* von Programmen, das allein schon aufgrund der hohen Datenmengen, die bei einem derartigen Prozeß bewegt werden müssen, eine technologische Herausforderung darstellt.
Nicht zuletzt muß auch für Parallelrechner - und damit unter erheblich erschwerten software-technischen Randbedingungen - organisatorisch sichergestellt werden, daß durch automatische Backup- und Archivierungssysteme leistungsfähige Mechanismen sowohl zur Datensicherung als auch zur Verwaltung großer Datenbestände zur Verfügung stehen [11].

4.3 Beratung und Ausbildung

Eine wesentliche Komponente im Dienstleistungsangebot von Rechenzentren ist die Beratung und Ausbildung. Dies gilt um so mehr für Parallelrechner, bei denen die Probleme - aufgrund der hohen Systemkomplexität und des geringen Reifegrades der Betriebs-Software - eher schwerer zu lösen sind. Es muß also für jeden wichtigen Software-Bereich Expertise gebildet und vorgehalten werden, um Anwender bei Fehlerfällen oder möglichen Performance-Problemen kurzfristig beraten und bei der Analyse der Problemursache unterstützen zu können. Dies kann in Einzelfällen auch bedeuten, daß Vorschläge zur Modifikation der mathematischen Methoden und Verfahren gemacht werden müssen, um zufriedenstellende Ergebnisse zu erzielen.
Der Umsetzung der im Beratungsprozeß gemachten Erfahrungen kommt dann im zweiten wichtigen Bereich - der Ausbildung der Anwender - eine ganz wesentliche Bedeutung zu. Während die erfahrenen Anwender gezielt Informationen über bekannte Software-Fehler und Performance-Optimierungen erwarten, müssen neue Anwender auf die zumeist hochkomplexe Systemarchitektur vorbereitet und geschult werden. Eine qualifizierte Beratung und Ausbildung dieser Art ist nur möglich, wenn die Mitarbeiter des Rechenzentrums vorab mit den Software-Komponenten gearbeitet und sich direkt mit den auftretenden Problemen auseinandergesetzt haben.

Die Umsetzung dieser Anforderungen hat im ZAM zur Formierung eines *Parallelrechner-Teams* geführt, in dem heute ca. 16 Mitarbeiter aus den verschiedenen Bereichen (Betriebssysteme, Anwendungsunterstützung, Programmierwerkzeuge, Spezial-Software) zu einer organisatorischen Einheit zusammengefaßt worden sind. Dieses Team sammelt bereits seit Jahresbeginn auf der am Konrad-Zuse-Zentrum in Berlin installierten CRAY T3D Erfahrungen bei der Parallelisierung und Portierung von Anwendungen auf diese neue CRAY-Plattform sammeln. Diese Mitarbeiter sind zum einen für die effektive Betreuung der Benutzer verantwortlich, behalten jedoch andererseits ihre Forschungs- und Entwicklungsaufgaben in ihrem Spezialgebiet. Gerade diese enge Kopplung von Forschungs- und Dienstleistungsaufgaben scheint bei der Lösung von Problemen in Anwendungen sehr hilfreich zu sein, da in vielen Fällen die Problemidentifikation und Einzelfallösung einhergeht mit der mittelfristigen Bereitstellung von Software-Komponenten, die solche Anwendungsproblem generell umgehen oder zumindest die Lokalisierung der Problemursache erleichtern.
Weiterhin hat sich gezeigt, daß für neue Benutzerprojekte in der Anfangsphase die feste Zuordnung eines Betreuers aus diesem Team, der die Erfahrungen aus anderen Projekten einfließen lassen kann, sehr hilfreich ist und die üblicherweise auftretenden Anfangsschwierigkeiten bei der Programmierung und Systemnutzung erheblich reduziert.

4.4 Entwicklung und Bereitstellung von Software-Werkzeugen

Bei der Programmierung von massiv-parallelen Rechnern zeigt sich immer wieder, daß mit realen Anwendungen oft nur ein Bruchteil der vom Hersteller angegebenen Peak-Leistung erreicht werden kann. Grund für diese Differenzen ist die im Vergleich zu ,,klassischen" Parallelrechnern mit physikalisch gemeinsamem Speicher deutlich komplexere Struktur dieser Systeme, die eine Optimierung der Anwendungen unter genauer Kenntnis der unterliegenden Maschinenarchitektur erfordert. Selbst wenn diese Kenntnisse vorhanden sind, ist es aber auch für einen erfahrenen Programmierer immer noch schwierig, die Auswirkungen von Programmänderungen auf den Programmablauf abzuschätzen bzw. nur anhand der Ausführungszeiten von Programmteilen Problemstellen zu identifizieren.
Profiling-Werkzeuge, wie sie von der sequentiellen Programmierung her bekannt sind und auch auf Parallelrechnern angeboten werden, geben mit ihren summarischen Angaben nur eine begrenzte Hilfestellung. Dies liegt daran, daß insbesondere für das auf massiv-parallelen Systemen weit verbreitete Programmiermodell des Message-Passing die zeitlichen Abhängigkeiten der einzelnen Instruktionsströme das Programmverhalten entscheidend beeinflussen. Deshalb ist Prozeß der Performance-Optimierung auf Parallelrechnern sehr aufwendig geworden, und entscheidende Laufzeitverbessungen lassen sich zumeist nur durch das detaillierte Verständnis der dynamischen Abläufe erzielen [12].
Allerdings wird diese Funktionalität nur sehr unzureichend von bisher verfügbaren Analysewerkzeugen wie *Paragraph* [13] oder *Pablo* [14] unterstützt. Da auch die Bereitstellung derartiger Werkzeugen durch die Hardware-Hersteller bisher immer

noch unzureichend ist, müssen leistungsfähige Software-Werkzeuge, die die effiziente Ausführung von parallelen Programmen auf parallelen Rechnern erleichtern oder sogar erst ermöglichen, häufig vor Ort entwickelt werden; dabei stehen nicht nur Hilfsmittel und Konzepte zur expliziten Programmierunterstützung im Vordergrund (z.B. *TOP*^{2} oder *SVM-Fortran* [1, 15, 5]), sondern auch Werkzeuge zum Test von Komponenten der System-Software, die für eine effektive Nutzung von Parallelrechnern unabdingbar sind (*PARBench*, [16]).

In diesem Beitrag soll etwas ausführlicher die Performance-Visualisierungs-umgebung *VAMPIR* erwähnt werden, die - auf der Basis des Software-Werkzeugs *PARvis [17, 18]* - im ZAM entwickelt wurde und nun auch den neuen Message-Passing-Standard *MPI* unterstützt. *VAMPIR* übersetzt eine die Trace-Datei eines zu untersuchenden Programmes in eine Menge von graphischen Darstellungen, zum Beispiel Zustandsdiagramme, Zeitlinien-Darstellungen und unterschiedlichste Arten von Statistiken. Darüber hinaus wird ein Animationsmodus unterstützt, der bei der Lokalisierung von Performance-Engpässen hilfreich sein kann, und es gibt flexible Filtermethoden, die es in einfacher Weise erlauben, die Menge der dargestellten Informationen geeignet einzuschränken [19].
Der interessanteste Teil von *VAMPIR* ist die leistungsfähige Zoom-Funktion, die es ermöglicht, Performance-Probleme auf jedem beliebigen Detaillierungsgrad zu identifizieren (siehe Abb. 9). Diese Eigenschaft erleichtert die Programmoptimierung, wodurch der Entwicklungszyklus einer Anwendung auf massiv-parallelen Rechnersystemen erheblich verkürzt werden kann.

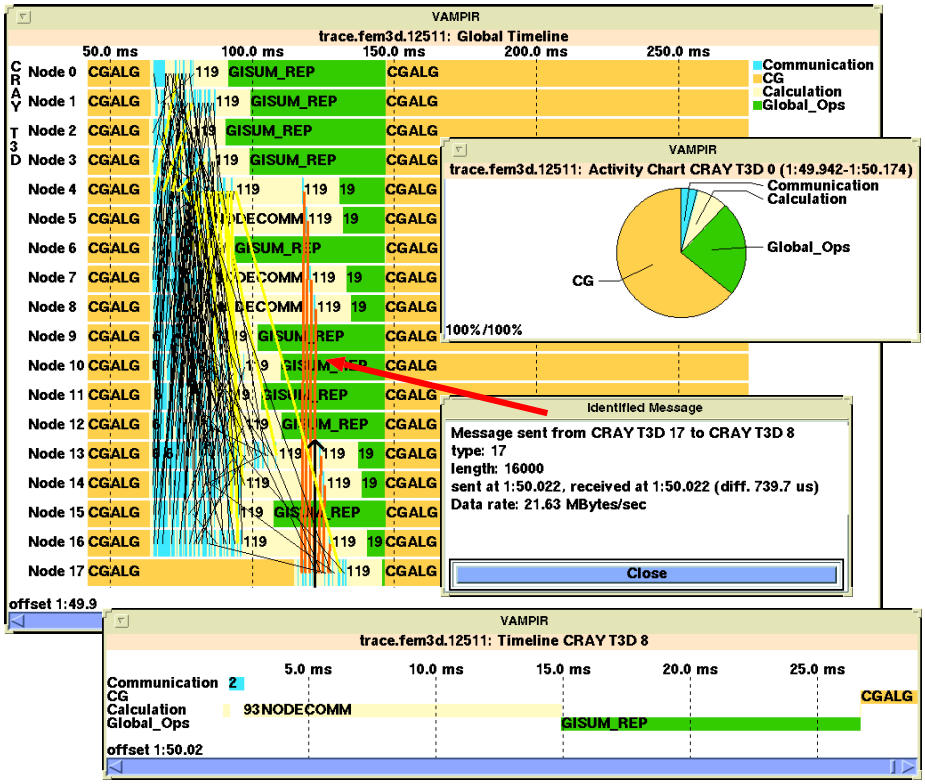

Abbildung 8
Verständnisgewinn durch leichtes Zooming

Wenn der Anwender - wie in Abb. 9 gezeigt - Informationen über den Ablauf seiner Unterprogramme erhalten will, muß eine Instrumentierungskomponente die Erzeugung einer entsprechenden Trace-Datei gewährleisten. Für den durch Fortran77 festgelegten Sprachumfang wurde - auf der Basis des ebenfalls im ZAM entwickelten Parsers PAFF [20] - das Instrumentierungswerkzeug *PARvis.inst* realisiert (siehe [18]) und für die Rechnerplattformen CRAY T3D, Intel Paragon und IBM SP 2 bereitgestellt. Im Rahmen einer Vereinbarung hat sich die Firma Cray bereiterklärt, für die neue Rechner-Plattform CRAY T3E ein Basis-Tool zu entwickeln, das die notwendigen Trace-Dateien sprachunabhängig - und damit für alle unterstützten Programmiersprachen - erstellt.

Durch die Portierung der Analyse-Umgebung *VAMPIR* auf verschiedene Hardware-Plattformen (unterstützt werden momentan die Rechnerlinien Sun, DEC Alpha,

IBM RS 6000, SGI und HP) und durch Funktionserweiterungen für den neuen Message-Passing Standard *MPI* steht nun ein universell einsetzbares Werkzeug zur Verfügung, das die dynamischen Vorgänge sichtbar macht und den Prozeß der Fehlersuche und der Performance-Optimierung gerade bei hochkomplexem Programmverhalten erheblich erleichtert.
Seit Anfang 1996 ist *VAMPIR* als kommerzielles Produkt der Firma PALLAS GmbH verfügbar; Produktinformationen können der Web-Seite *http://www.pallas.de* entnommen werden.

5 Ausblick

Für die Lösung von komplexen Problemen - den sogenannten *Grand Challenges* - durch Simulationsrechnungen stellt der im Forschungszentrum Jülich ab Juli 1996 verfügbare Parallelrechner CRAY T3E einen großen Schritt dar, und er wird sowohl für die Forschungslandschaft Jülich als auch bundesweit für die Projekte des Höchstleistungsrechenzentrums HLRZ zu einem wichtigen Forschungsinstrument werden. Die Erwartungshaltung ist nach den positiven Erfahrungen mit dem Vorgängermodell CRAY T3D hoch und wird, wenn man die bisherigen festen Bestellungen dieses Rechners - zu einem wesentlichen Teil auch aus Deutschland - betrachtet, von vielen Experten geteilt. Es bleibt zu hoffen, daß auch diesmal die hohen Erwartungen bei Auslieferung der Systeme voll erfüllt werden.
Unsere bisherigen Erfahrungen mit Parallelrechner haben jedoch gezeigt, daß die effiziente Nutzung - neben der Stabilität der System-Software und möglichst hohen Werten für die real erzielbare Performance - auch von der Leistungsfähigkeit des Dienstleistungsangebotes eines Rechenzentrums abhängt. Neben den klassischen Aufgaben muß hier - als neue und wichtige Teilkomponente des Aufgabenspektrums - die Entwicklung und Bereitstellung von Software-Werkzeugen sowohl zur Unterstützung der Programmierung als auch zur Performance-Analyse und Optimierung genannt werden, die für diese neue Klasse von Rechnern eine immer stärkere Bedeutung gewinnt.

Dank

Ich möchte mich bei Dr. Wilfried Oed und Dr. Reiner Vogelsang herzlich für die fachliche Unterstützung bei der Erstellung dieses Beitrages und darüber hinaus für ihre Bereitschaft zu zahlreichen fruchtbaren Diskussionen bedanken.

Literatur

1. F. Hoßfeld and W.E. Nagel, Per aspera ad Astra - On the Way to Parallel Processing, in *Proc. Seminar „Supercomputing '95"*, K.G. Saur Verlag, München (1995), pp. 246-259.

2. W.E. Nagel, Effektive Nutzung von Parallelrechnern in Rechenzentrumsumgebungen: Eine Herausforderung an das Dienstleistungsangebot, In: D. Wall (Hrsg.): *Organisation und Betrieb von DV-Versorgungssystemen*, Deutscher Universitätsverlag, Wiesbaden (1995), S. 189-200
3. S. Reinhardt, The CRAY T3E System, In: Proc. Spring 1996 Cray Users Group Meeting, to appear.
4. W. Oed. Massiv-paralleles Prozessorsystem CRAY T3E, Technische Dokumentation, Cray Research GmbH, München (1996).
5. M. Gerndt and R. Berrendorf, Parallelizing Applications with SVM-Fortran, In Proc. HPCN Europe 1995, LNCS 919, Mailand, Italien (1995), pp. 793-798.
6. CRAY MPP Fortran Reference Manual, SR-2504 Rel. 6.1 (1994).
7. T. MacDonald, The HPF-CRAFT Programming Model, In: Proc. Spring 1996 Cray Users Group Meeting, to appear.
8. J. Docter, Integration eines massiv-parallelen Rechners in die Produktionsumgebung eines Rechenzentrums, in *Verteilte Systeme - Organisation und Betrieb* (Löw, H.-P., Partosch, G., Hrsg.), pp. 94-102, DUV Deutscher-UniversitätsVerlag, Wiesbaden (1993).
9. W. E. Nagel, Ein verteiltes Scheduler-System für Mehrprozessorrechner mit gemeinsamem Speicher: Untersuchungen zur Ablaufplanung von parallelen Programmen, Berichte des Forschungszentrums Jülich Jül-2850 (1993).
10. F. Przybylski, Scheduling-Verfahren für Rechner mit verteiltem Speicher: Eine vergleichende Übersicht, Berichte des Forschungszentrums Jülich Jül-2598 (1992).
11. L. Wollschläger, Zentrale Datensicherung bei dezentraler Datenhaltung, in *Verteilte Systeme - Organisation und Betrieb* (Löw, H.-P., Partosch, G., Hrsg.), pp. 155-165, Deutscher UniversitätsVerlag, Wiesbaden (1993).
12. A. Arnold, J. Bernert, W.E. Nagel, and M. Röth, Performance-Analyse paralleler Programme: Die PARvis-Visualisierungsumgebung}, In: PARS-Mitteilungen Nr. 14,1995, pp. 191-200.
13. Paragon Application Tools Users Guide, Intel Corporation (1993).
14. D.A. Reed, R.A. Aydt, T.M. Madhyastha, R.J. Noe, K.A. Shields, and B.W. Schwartz, An Overview of the Pablo Performance Analysis Environment, Technical Report, Dept. of Computer Science, University of Illinois, Urbana-Champaign (1992).
15. U. Detert and M. Gerndt, TOP^2 - Tool Suite for the Development and Testing of Parallel Applications, Proc. CONPAR'94/VAPP VI, Universität Linz (1994), pp. 196-207.
16. W.E. Nagel and M.A. Linn, Benchmarking Parallel Programs in a Multiprogramming Environment: The PARbench System, In: Advances in Parallel Computing, Vol. 8: *Computer Benchmarks* (Dongarra, J., Gentzsch, W., eds.) pp. 302-322, Elsevier Science Publishers B.V. (1993).
17. W.E. Nagel and A. Arnold, Performance Visualization of Parallel Programs - The PARvis Environment, in *Proc. 1994 Intel Supercomputer Users Group (ISUG) Conference*, San Diego, USA, pp. 24-31.

18. A. Arnold, U. Detert, and W.E. Nagel, Performance Optimization of Parallel Programs: Tracing, Zooming, Understanding, In: Proc. Spring 1995 Cray Users Group Meeting, pp. 252--258.
19. W.E. Nagel, A. Arnold, M. Weber, H.Ch. Hoppe, and K. Solchenbach, VAMPIR: Visualization and Analysis of MPI Resources, In: Supercomputer Vol. XII (No. 1) (1996), pp. 69-80.
20. R. Berrendorf, Der FORTRAN-Parser PAFF als wiederverwendbares Modul für Programmier-Tools, Forschungszentrum Jülich (KFA), Jül-Spez-537 (1989).